CROSS-COVARIANCE ESTIMATION FOR EKF-BASED INERTIAL AIDED MONOCULAR SLAM

Markus Kleinert[a] **and Uwe Stilla**[b]

[a]Fraunhofer IOSB, Department Scene Analysis, Gutleuthausstr. 1, 76275 Ettlingen, markus.kleinert@iosb.fraunhofer.de
[b]Technische Universität München, Photogrammetry and Remote Sensing, Arcisstrasse 21, 80333 München, stilla@bv.tum.de

KEY WORDS: Monocular SLAM, Inertial measurement system, Extended Kalman filter, Correlations, Estimation

ABSTRACT:

Repeated observation of several characteristically textured surface elements allows the reconstruction of the camera trajectory and a sparse point cloud which is often referred to as "map". The extended Kalman filter (EKF) is a popular method to address this problem, especially if real-time constraints have to be met. Inertial measurements as well as a parameterization of the state vector that conforms better to the linearity assumptions made by the EKF may be employed to reduce the impact of linearization errors. Therefore, we adopt an inertial-aided monocular SLAM approach where landmarks are parameterized in inverse depth w.r.t. the coordinate system in which they were observed for the first time. In this work we present a method to estimate the cross-covariances between landmarks which are introduced in the EKF state vector for the first time and the old filter state that can be applied in the special case at hand where each landmark is parameterized w.r.t. an individual coordinate system.

1 INTRODUCTION

1.1 Motivation

Navigation in unknown environments is often hindered by the absence of external positioning information. In the context of pedestrian navigation for instance, the GPS signal may be temporarily lost or severely disturbed due to multipath effects in urban canyons. The need to cope with such situations has motivated the research in systems which are capable of building and maintaining a map of the environment while at the same time localizing themselves w.r.t. that map. This problem is commonly referred to as simultaneous localization and mapping (SLAM). For the particular application of pedestrian navigation, a promising approach is to combine a low-cost inertial measurement unit (IMU) and a camera to an inertial aided monocular SLAM system and perform sensor data fusion in an extended Kalman filter (EKF), *cf.* (Veth and Raquet, 2006). Here, characteristically textured surface elements serve as landmarks which can be observed by the camera to build up a map while the IMU's acceleration and angular rate measurements are integrated in order to obtain a short-time accurate prediction of the camera's pose and thereby help to reduce linearization error.

1.2 Related Work

An important aspect of such monocular SLAM systems is the representation of landmarks in the filter state vector. Montiel *et al.* have proposed an inverse depth parameterization of landmarks that conforms well to the linearity assumptions made by the EKF and offers the possibility of instantly introducing new landmarks in the filter state with only one observation (Montiel et al., 2006). In the original inverse depth parameterization six additional parameters are included in the filter state for each freshly introduced landmark. To alleviate the computational burden imposed by this over-parameterization, Civera *et al.* introduce a method to transform landmarks to Cartesian coordinates once their associated covariance has sufficiently collapsed (Civera et al., 2007). Alternatively, Pietzsch proposes to initialize bundles of landmarks and to estimate only the inverse distance to the origin of the coordinate system of the camera that observed the landmarks for the first time for each landmark individually and the position and orientation of the camera coordinate system for the whole bundle (Pietzsch, 2008).

The importance of the cross-covariance terms in SLAM is stressed in a seminal work by Dissanayake *et al.* (Dissanayake et al., 2001). Julier and Uhlmann present a detailed investigation of the consistency of EKF-SLAM implementations (Julier and Uhlmann, 2001). Therein it is shown that errors in the estimated cross-covariance terms due to linearization errors lead to inconsistent estimates. A comparison of several landmark parameterizations for monocular SLAM regarding their effects on filter consistency is provided by Solà (Solà, 2010). This work also gives a detailed description of landmark initialization in monocular SLAM.

1.3 Contribution

Our approach is to parameterize each landmark in inverse depth polar coordinates w.r.t. the coordinate system of the camera at the time of its first observation. Therefore, the camera's orientation and position as well as the parameters that describe the landmark's position in the camera coordinate frame have to be stored for each landmark. However, we regard the camera's position and orientation as fix model parameters and thus only include the three parameters which describe the landmark's uncertain position in the filter state, thereby avoiding overparameterizing the landmark's position. Since the camera's position and orientation are regarded as fix model parameters, the corresponding uncertainty estimate has to be conveyed to the landmark's uncertainty estimate. In addition, the cross-covariances between the new landmark and the landmarks already present in the filter state have to be computed. This is aggravated by the fact, that the landmark coordinates in the filter state are given with respect to distinct coordinate frames, which precludes the adaption of standard error propagation methods in this case. The main contribution of our work is a method to convey the uncertainty estimate from the camera to the landmark parameters and to determine the cross-covariances between the new landmark and the parameters already present in the filter state.

2 EKF-SLAM FORMULATION

This section describes the EKF-SLAM approach employed in this work with an emphasis on the parameterization of landmarks.

2.1 Coordinate systems

The following coordinate systems are of particular interest for the derivation of the cross-covariances in sec. 2.5.2. An overview is also given in fig. 1.

The body or IMU-coordinate system $\{b\}$ that is aligned to the IMU's axes and therefore describes position and orientation of the whole sensor system. Its position and orientation are included in the filter state.

The camera coordinate system $\{c\}$. The camera's position and orientation are not part of the filter state. They can be calculated from the IMU's position and orientation by means of the camera-IMU transformation that only depends on the mechanical setup and is assumed to be fix and known in this work.

The navigation frame $\{n\}$. This is the reference frame whose x- and y- axis point north- and eastwards while its z-axis points in the direction of local gravity. We assume that the distance to the local reference frame is small compared to the curvature of the earth and therefore the direction of gravity can be considered constant during operation of the system. In this case the position of the navigation frame can be chosen arbitrarily.

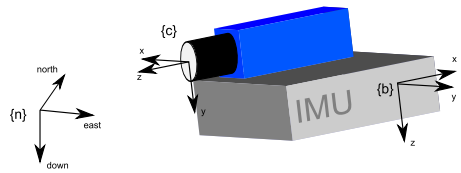


Figure 1: Overview of the coordinate systems used in this work

2.2 State parameterization

The goal is to determine the position and orientation of the body frame w.r.t. the navigation frame and a sparse map of point landmarks. Hence, the EKF state vector comprises parameters which describe the IMU's motion and biases as well as the coordinates of observed landmarks:

$$\mathbf{s}_t = \left[\underbrace{{}^n\mathbf{p}_b^{\mathrm{T}}\ {}^n\mathbf{v}_b^{\mathrm{T}}\ \mathbf{b}_a^{\mathrm{T}}\ \mathbf{q}_b^{n\mathrm{T}}\ \mathbf{b}_g^{\mathrm{T}}}_{\mathbf{s}'} \underbrace{\mathbf{Y}_1^{\mathrm{T}} \ \dots \ \mathbf{Y}_N^{\mathrm{T}}}_{\mathbf{m}} \right]^{\mathrm{T}} \tag{1}$$

Where ${}^n\mathbf{p}_b$ and ${}^n\mathbf{v}_b$ are the IMU's position and velocity w.r.t. the navigation frame, the unit quaternion $\mathbf{q}_b^n$ represents the orientation and $\mathbf{b}_a$, $\mathbf{b}_g$ are the sensor biases, which systematically disturb acceleration and angular rate measurements. For convenience, the landmark coordinates $\mathbf{Y}_i$ are subsumed in the map vector $\mathbf{m}$. Similarly, the part of the state vector that describes the motion of the IMU is denoted by $\mathbf{s}'$. In the following, estimated values are denoted with by a hat $(\hat{\cdot})$ and a tilde $(\tilde{\cdot})$ is used to indicate the error, *i.e.* the deviation between a true value $(\cdot)$ and its estimate: $(\tilde{\cdot}) = (\cdot) - (\hat{\cdot})$.

2.2.1 Error state formulation.

Since the EKF relies on a truncation of the Taylor series expansion of the measurement equation as well as the time update step after the first derivative, it can be regarded as an estimator for the state error $\tilde{\mathbf{s}}$. This is the basis for the error state formulation of the EKF which is commonly used for GPS-INS integration, *cf.* (Farrell and Barth, 1999, pp. 199-222). Therefore, the covariance matrix associated with the filter state describes the distribution of $\tilde{\mathbf{s}}$ under the assumption that the errors follow a normal distribution. It is given by

$$P = \begin{bmatrix} P_{\tilde{\mathbf{s}}',\tilde{\mathbf{s}}'} & P_{\tilde{\mathbf{s}}',\tilde{\mathbf{m}}} \\ P_{\tilde{\mathbf{m}},\tilde{\mathbf{s}}'} & P_{\tilde{\mathbf{m}},\tilde{\mathbf{m}}} \end{bmatrix} . \tag{2}$$

The error of the estimated orientation can be written in terms of the incremental orientation that aligns the estimated coordinate system with the unknown true coordinate system:

$$\mathbf{q}_b^n = \mathbf{q}(\Psi) * \hat{\mathbf{q}}_b^n \quad , \quad \mathbf{q}(\Psi) \approx \left[1 \ \tfrac{\Psi}{2}^{\mathrm{T}}\right]^{\mathrm{T}} \tag{3}$$

Where $*$ denotes quaternion multiplication.

2.2.2 Landmark parameterization.

The coordinate vector of the i-th landmark in the filter state $\mathbf{Y}_i$ describes the position of the landmark in inverse depth polar coordinates w.r.t. the coordinate frame $\{c_k\}$ of the camera at the time when the landmark was observed for the first time as illustrated in fig. 2.

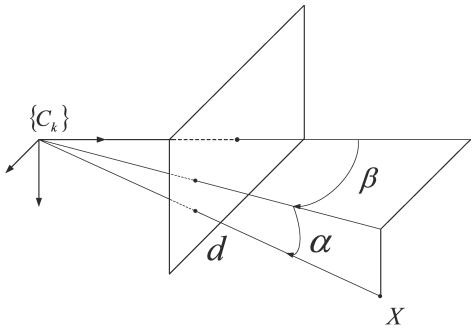


Figure 2: Landmark parameterization in inverse depth polar coordinates. X is the Cartesian coordinate vector associated with the inverse depth parameterization $Y = [\alpha\ \beta\ \rho\ ]$, with elevation angle α, azimuth β and inverse depth $\rho = 1/d$, all w.r.t. the anchor system $\{c_k\}$.

Using the camera coordinate frame $\{c_k\}$ as an anchor for the landmark therefore avoids over-parameterization in the filter state and should thus increase computational efficiency and stability during Kalman filter updates. In order to determine the position of a landmark in the reference coordinate system, the transformation from the anchor coordinate system to the reference frame is needed:

$${}^n\mathbf{X} = C_{c_k}^n \cdot \frac{1}{\rho} \underbrace{\begin{bmatrix} \cos(\alpha)\sin(\beta) \\ \sin(\alpha) \\ \cos(\alpha)\cos(\beta) \end{bmatrix}}_{\overline{\mathbf{Y}}(\mathbf{Y})} + {}^n\mathbf{p}_{c_k} \tag{4}$$

In the above equation, the direction cosine matrix $C_{c_k}^n$ describes the orientation of of the anchor system and ${}^n\mathbf{p}_{c_k}$ is its position. $\overline{\mathbf{Y}}$ is a unit vector that points in the direction of the projection ray.

For every landmark $C^n_{c_k}$ and ${}^n\mathbf{p}_{c_k}$ are therefore stored in a separate data structure because they are not part of the filter state although they vary for different landmarks.

2.3 Time update

2.3.1 Integration of inertial measurements. During the time update, the estimated state is propagated by integrating the inertial measurements. To integrate angular rate measurements ω, a quaternion that describes the incremental rotation in the body frame is formed from the angular rate measurements and subsequently used to update the orientation estimate:

$$\begin{aligned} \hat{\omega} &= \omega - \hat{\mathbf{b}}_g \\ \hat{\mathbf{q}}^n_{b,t+\tau} &= \hat{\mathbf{q}}^n_{b,t} * \mathbf{q}(\hat{\omega}) \end{aligned} \tag{5}$$

Where τ is the time interval between two consecutive inertial measurements. Acceleration measurements have to be transformed to the reference coordinate system before the gravitational acceleration can be subtracted. Then, the resulting estimate of acceleration is used to integrate velocity and position:

$$\begin{aligned} {}^n\hat{\mathbf{a}}_{b,t} &= C(\hat{\mathbf{q}}^n_{b,t}) \cdot ({}^b\tilde{\mathbf{a}}_{b,t} - \hat{\mathbf{b}}_{a,t}) + {}^n\mathbf{g} \\ {}^n\hat{\mathbf{p}}_{b,t+\tau} &= {}^n\hat{\mathbf{p}}_{b,t} + {}^n\hat{\mathbf{v}}_{b,t} \cdot \tau + \tfrac{1}{2}{}^n\hat{\mathbf{a}}_{b,t} \cdot \tau^2 \\ {}^n\hat{\mathbf{v}}_{b,t+\tau} &= {}^n\hat{\mathbf{v}}_{b,t} + {}^n\hat{\mathbf{a}}_{b,t} \cdot \tau \end{aligned} \tag{6}$$

2.3.2 Covariance propagation. The physical model described by equations 5-6 corresponds to the first order differential equation that describes the propagation of the estimation error $\tilde{\mathbf{s}}'$ for the time dependent part of the state vector:

$$\dot{\tilde{\mathbf{s}}}' = F' \cdot \tilde{\mathbf{s}}' + G' \cdot \mathbf{n} \tag{7}$$

Here, F' is determined by the physical model and $\mathbf{n}$ summarizes the white noise terms. From F' the discrete time transition matrix Φ' is computed and used thereafter to calculate the propagated error state covariance matrix $P_{t+\tau}$ as stated below:

$$\Phi' = \exp(F' \cdot \tau) \approx I_{15\times 15} + F' \cdot \tau \tag{8}$$

$$P'_{t+\tau} = \Phi' \cdot P_{\tilde{\mathbf{s}}'_t, \tilde{\mathbf{s}}'_t} \cdot \Phi'^{\mathrm{T}} + \Phi' \cdot G' \cdot Q \cdot G'^{\mathrm{T}} \cdot \Phi'^{\mathrm{T}} \cdot \tau \tag{9}$$

$$P_{t+\tau} = \begin{bmatrix} P'_{t+\tau} & \Phi' \cdot P_{\tilde{\mathbf{s}}'_t, \tilde{\mathbf{m}}_t} \\ P_{\tilde{\mathbf{m}}_t, \tilde{\mathbf{s}}'_t} \cdot \Phi'^{\mathrm{T}} & P_{\tilde{\mathbf{m}}_t, \tilde{\mathbf{m}}_t} \end{bmatrix} \tag{10}$$

In the expression above, Q is the power spectral density matrix which characterizes the noise vector $\mathbf{n}$.

2.4 Measurement update

New images are continuously triggered by the IMU as it moves along its trajectory. Therein the image coordinates of point features are extracted and tracked. The coordinates of all features extracted in one image are stacked together to form the measurement vector that is subsequently used to update the state vector.

2.4.1 Landmark observation model. The observation model describes the relationship between the observed image coordinates and the state vector entries. For this purpose, the estimate of each observed landmark in inverse depth polar coordinates w.r.t. its anchor system is first transformed to Cartesian coordinates w.r.t. the navigation frame as shown in eq. 4. Subsequently, the coordinates are transformed to the current camera coordinate system and projected on the image plane:

$$\begin{aligned} \mathbf{z} &= \mathbf{h}(\mathbf{s}) + \mathbf{v} \\ &= \pi(C^c_n \cdot (\frac{1}{\rho} \cdot C^n_{c_k} \cdot \overline{\mathbf{Y}}(\alpha,\, \beta) + {}^n\mathbf{p}_{c_k} - {}^n\mathbf{p}_c)) + \mathbf{v} \\ &= \pi(\underbrace{C^c_n \cdot (C^n_{c_k} \cdot \overline{\mathbf{Y}}(\alpha,\, \beta) + \rho \cdot ({}^n\mathbf{p}_{c_k} - {}^n\mathbf{p}_b)) + \rho \cdot {}^c\mathbf{p}_b}_{{}^c\mathbf{X}}) \\ &\quad + \mathbf{v} \end{aligned} \tag{11}$$

Where $\mathbf{z}$ are the measured image coordinates, $\mathbf{v}$ is the zero mean measurement noise, and $\pi(\cdot)$ is the projection function. Eq. 11 describes the projection of one landmark. The Jacobian of $\mathbf{h}(\mathbf{s})$ w.r.t. the entries of the state vector for landmark no. i is:

$$H_i = J_\pi \begin{bmatrix} J_{\mathbf{p}}\ 0_{3\times 6}\ J_\Psi\ 0_{3\times 3}\ 0_{3\times 3\cdot(i-1)}\ J_{\mathbf{Y}}\ 0_{3\times 3\cdot(N-i)} \end{bmatrix} . \tag{12}$$

With the derivatives J_Ψ, $J_{\mathbf{p}}$, and $J_{\mathbf{Y}}$ of ${}^c\mathbf{X}$ w.r.t. the orientation $\mathbf{q}^n_b$, the position ${}^n\mathbf{p}_b$, and the landmark $\mathbf{Y}$. Similarly to the measurement vector, the Jacobian H for the whole set of measurements is obtained by stacking the measurement Jacobians for individual landmarks.

Given the measurement model derived in this section and the prediction from sec. 2.3, an EKF update step can be performed as described in (Bar-Shalom et al., 2001, pp. 200-217).

2.5 Introduction of new landmarks

Landmarks are deleted from the filter state if they could not be observed in a predefined number of consecutive images. Whenever the number of landmarks in the state drops below a predetermined threshold, new landmarks have to be introduced. Since the standard formulation of the Kalman filter does not allow for a variable state size, the new filter state entries have to be estimated based on previous observations. The inverse depth polar coordinates for each new feature can be calculated based on the image coordinates of its last observation and the camera calibration by means of trigonometric functions:

$$\mathbf{Y}_{new} = f(p_x, p_y, k, \rho_{init}) \tag{13}$$

Where p_x, p_y are the measured image coordinates, k contains the intrinsic camera calibration parameters, ρ_{init} is an arbitrarily chosen inverse depth measurement and $f(\cdot)$ is the back projection function, which projects to a point on the projection ray through the observed image coordinates. In the following, J_f denotes the Jacobian of f w.r.t. p_x, p_y ,and ρ_{init}.

A new landmark is introduced in the Kalman filter state by augmenting the state vector with the initial estimate of the landmark's position $\mathbf{Y}_{new}$:

$$\mathbf{s}_{new} = \begin{bmatrix} \mathbf{s}'^{\mathrm{T}}\ \mathbf{m}^{\mathrm{T}}\ \mathbf{Y}^{\mathrm{T}}_{new} \end{bmatrix}^{\mathrm{T}} \tag{14}$$

In addition the covariance matrix has to be augmented with the new cross-covariance terms and the covariance of the new landmark:

$$P_{new} = \begin{bmatrix} P_{\tilde{\mathbf{s}}',\tilde{\mathbf{s}}'} & P_{\tilde{\mathbf{s}}',\tilde{\mathbf{m}}} & P_{\tilde{\mathbf{s}}',\tilde{\mathbf{Y}}_{new}} \\ P_{\tilde{\mathbf{m}},\tilde{\mathbf{s}}'} & P_{\tilde{\mathbf{m}},\tilde{\mathbf{m}}} & P_{\tilde{\mathbf{m}},\tilde{\mathbf{Y}}_{new}} \\ P_{\tilde{\mathbf{Y}}_{new},\tilde{\mathbf{s}}'} & P_{\tilde{\mathbf{Y}}_{new},\tilde{\mathbf{m}}} & P_{\tilde{\mathbf{Y}}_{new},\tilde{\mathbf{Y}}_{new}} \end{bmatrix} \quad (15)$$

2.5.1 Conventional approach. A commonly used method to calculate the initial landmark estimate and the associated covariance entries is to define an inverse observation function

$$\mathbf{Y}_{new} = \mathbf{g}(\mathbf{s}', \mathbf{z}, \rho) \quad (16)$$

that depends on one or more measurements $\mathbf{z}$, the sensor system's position and orientation in $\mathbf{s}'$ as well as on predefined parameters like ρ. Let $J_{s'}$, J_z, and J_ρ be the derivatives of $\mathbf{g}(\cdot)$ w.r.t. $\mathbf{s}'$, $\mathbf{z}$, and ρ. The sought-after covariance entries can then be approximated as follows (Solà, 2010):

$$P_{\tilde{\mathbf{Y}}_{new},\tilde{\mathbf{s}}'} = J_{s'} P_{\tilde{\mathbf{s}}',\tilde{\mathbf{s}}'} \quad (17)$$
$$P_{\tilde{\mathbf{Y}}_{new},\tilde{\mathbf{m}}} = J_{s'} P_{\tilde{\mathbf{s}}',\tilde{\mathbf{m}}'} \quad (18)$$
$$P_{\tilde{\mathbf{Y}}_{new},\tilde{\mathbf{Y}}_{new}} = J_{s'} P_{\tilde{\mathbf{s}}',\tilde{\mathbf{s}}'} J_{s'}^{\mathrm{T}} + J_z R J_z^{\mathrm{T}} + J_\rho \sigma_\rho^2 J_\rho^{\mathrm{T}} \quad (19)$$

Where σ_ρ^2 is the variance of the initial inverse depth estimate and R is the measurement noise covariance matrix.

2.5.2 Proposed method. The scheme presented in the previous section is not directly applicable to landmark parameterizations as described in sec. 2.2.2. In this case function $f(\cdot)$ from eq. 13 takes the role of $g(\cdot)$ in the above equations. The problem is, that $f(\cdot)$ does not depend on the system's position and orientation. Thus, the Jacobian $J_{s'}$ and with it the cross-covariances $P_{\tilde{\mathbf{Y}}_{new},\tilde{\mathbf{s}}'}$, $P_{\tilde{\mathbf{Y}}_{new},\tilde{\mathbf{m}}}$ become zero. As a result, the uncertainty of the position and orientation estimate of the sensor system will be neglected when eq. 17- 19 are applied.

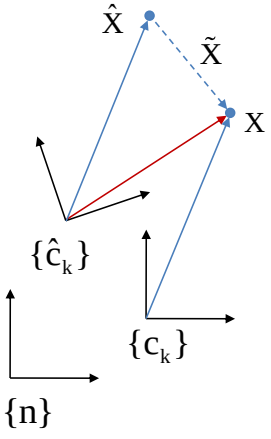


Figure 3: Position of a feature expressed in the true camera coordinate system c_k and the estimated camera coordinate system $\hat{c}_k$.

Fig. 3 depicts the situation when a landmark's position estimate in Cartesian coordinates w.r.t. the reference frame $\{n\}$ is computed based on an erroneous estimate $\{\hat{c}_k\}$ of the camera's position and orientation. The blue arrows indicate the landmark position estimates in the true camera coordinate system $\{c_k\}$ and in the estimated camera coordinate system $\{\hat{c}_k\}$ that are consistent with the observed feature locations. Note, that the vectors $\hat{\mathbf{X}}$, $\mathbf{X}$ expressed in camera coordinates are identical in case of a perfect measurement. The red arrow marks the correct landmark estimate in the erroneously estimated camera coordinate system $\{c_k\}$. The key idea behind our approach is to express the landmark position error $\tilde{\mathbf{X}} = \mathbf{X} - \hat{\mathbf{X}}$ (the dashed arrow) in the estimated camera coordinate frame in terms of the unknown transformation between the true and the estimated camera coordinate systems and to use this transformation to propagate the error from the camera coordinate system estimate to the landmark position estimate and to calculate the cross-covariance entries.

In the ensuing derivation, the orientation error model described in sec. 2.2.1 will be used. The transformation between the true coordinate system $\{c_k\}$ and the estimated coordinate system $\{\hat{c}_k\}$ can be written in terms of a rotation matrix $C_{c_k}^{\hat{c}_k}$ and the position ${}^{\hat{c}_k}\mathbf{p}_{c_k}$. The rotation matrix $C_{c_k}^{\hat{c}_k}$ depends on the Rodrigues vector that is defined in eq. 3:

$$C_{c_k}^{\hat{c}_k} = C(\Psi_{c_k}^{\hat{c}_k}) \approx (I - \left\lfloor \Psi_{c_k}^{\hat{c}_k} \right\rfloor_\times) \quad (20)$$
$$\Psi_{c_k}^{\hat{c}_k} = \hat{C}_n^{c_k} \cdot \Psi \quad (21)$$

Where $\hat{C}_n^{c_k} = C_n^{\hat{c}_k} = C_{\hat{n}}^{c_k}$ is the rotation matrix calculated from the estimated orientation of the sensor system and the IMU-camera calibration and $\Psi_{c_k}^{\hat{c}_k}$ is the orientation error expressed in the estimated camera coordinate system. With this an expression for the landmark error in Cartesian coordinates can be derived where the index k, which is used to mark the anchor coordinate system for a landmark, is omitted for brevity:

$$\begin{aligned} {}^{\hat{c}}\tilde{\mathbf{X}} &= {}^{\hat{c}}\mathbf{X} - {}^{\hat{c}}\hat{\mathbf{X}} \\ &= C_c^{\hat{c}} \cdot {}^{c}\mathbf{X} + {}^{\hat{c}}\mathbf{p}_c - {}^{\hat{c}}\hat{\mathbf{X}} \\ &= C_c^{\hat{c}} \cdot {}^{c}\mathbf{X} + C_n^{\hat{c}} \cdot ({}^{n}\mathbf{p}_b - {}^{n}\mathbf{p}_{\hat{b}}) + {}^{c}\mathbf{p}_b - C_c^{\hat{c}} \cdot {}^{c}\mathbf{p}_b - {}^{\hat{c}}\hat{\mathbf{X}} \\ &= C_c^{\hat{c}} \cdot ({}^{c}\mathbf{X} - {}^{c}\mathbf{p}_b) - {}^{\hat{c}}\hat{\mathbf{X}} + {}^{c}\mathbf{p}_b + C_n^{\hat{c}} \cdot ({}^{n}\mathbf{p}_b - {}^{n}\mathbf{p}_{\hat{b}}) \\ &\approx C_c^{\hat{c}} \cdot ({}^{\hat{c}}\hat{\mathbf{X}} - {}^{c}\mathbf{p}_b) - {}^{\hat{c}}\hat{\mathbf{X}} + {}^{c}\mathbf{p}_b + C_n^{\hat{c}} \cdot ({}^{n}\mathbf{p}_b - {}^{n}\mathbf{p}_{\hat{b}}) \end{aligned} \quad (22)$$

In eq. 22 the approximation ${}^{c}\mathbf{X} \approx {}^{\hat{c}}\hat{\mathbf{X}}$ is used, thereby assuming that the main error is caused by the erroneous estimate of the coordinate system, *cf.* fig. 3. Using the small angle approximation from eq. 20, ${}^{\hat{c}}\tilde{\mathbf{X}}$ can be written as a linear function of the errors of the estimated state:

$$\begin{aligned} {}^{\hat{c}}\tilde{\mathbf{X}} &= (I + \left\lfloor \Psi_{c_k}^{\hat{c}_k} \right\rfloor_\times) \cdot ({}^{\hat{c}}\hat{\mathbf{X}} - {}^{c}\mathbf{p}_b) - {}^{\hat{c}}\hat{\mathbf{X}} + {}^{c}\mathbf{p}_b + C_n^{\hat{c}} \cdot ({}^{n}\mathbf{p}_b - {}^{n}\mathbf{p}_{\hat{b}}) \\ &= \left\lfloor {}^{c}\mathbf{p}_b - {}^{\hat{c}}\hat{\mathbf{X}} \right\rfloor_\times \cdot C_n^{\hat{c}} \cdot \Psi + C_n^{\hat{c}} \cdot {}^{n}\tilde{\mathbf{p}}_b \end{aligned} \quad (23)$$

This is the sought relationship between the errors of the current orientation and position estimates for the sensor system and the error of the newly introduced landmark in Cartesian coordinates w.r.t. the current camera coordinate system. It only depends on entities, which are either estimated or known a-priori, like the the position of the camera in the body coordinate system. The partial derivatives of the landmark coordinates w.r.t. the IMU's position and orientation follow directly from eq. 23:

$$\frac{\partial^{c_k}\mathbf{X}}{\partial\Psi} = \left[{}^{c}\mathbf{p}_b - {}^{\hat{c}}\hat{\mathbf{X}}\right]_\times \cdot C_n^{\hat{c}} \tag{24}$$

$$\frac{\partial^{c_k}\mathbf{X}}{\partial^n\mathbf{p}_b} = \hat{C}_n^{\hat{c}} \tag{25}$$

Given the partial derivatives 24-25, a new landmark can be introduced to the filter state by following the subsequent steps:

1. Calculate the inverse depth polar coordinates $\mathbf{Y}$ and the Cartesian coordinates ${}^{c_k}\mathbf{X}$ for the new landmark given the observation and an arbitrarily chosen inverse depth estimate ρ.

2. Calculate the partial derivative $\partial\mathbf{Y}/\partial^{c_k}\mathbf{X}$, which describes how the inverse depth polar coordinates vary with ${}^{c_k}\mathbf{X}$.

3. Determine $J_{s'}$:

$$J_{s'} = \frac{\partial\mathbf{Y}}{\partial^{c_k}\mathbf{X}} \cdot \left[\frac{\partial^{c_k}\mathbf{X}}{\partial^n\mathbf{p}_b}\ 0_{3\times6}\ \frac{\partial^{c_k}\mathbf{X}}{\partial\Psi}\ 0_{3\times3}\right] \tag{26}$$

4. Use eqs. 17-19 to calculate the missing covariance matrix entries and augment the filter state according to eqs. 14 and 15.

3 RESULTS AND DISCUSSION

3.1 Experimental setup

The cross covariance estimation algorithm derived in sec. 2.5.2 was compared against a naive landmark introduction method that simply discards the cross-correlations in a number of simulation runs. For the simulation a reference trajectory was defined by two C^2 splines. One spline determines the viewing direction while the other spline describes the position of the IMU. The second derivative of this spline provides acceleration measurements whereas angular rate measurements are derived from the differential rotation between two sampling points. In addition, image measurements were generated by projecting landmark positions onto the image plane.

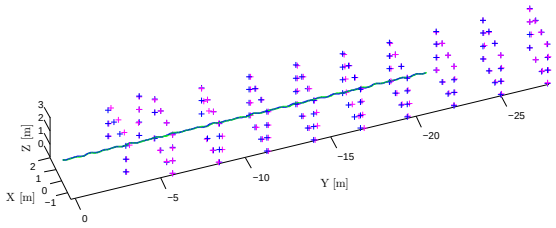
Artificial white Gaussian noise was added to all measurements. Its variance was chosen to resemble the characteristics of a good tactical grade IMU with $0.47^\circ/\sqrt{\text{h}}$ angular random walk and $0.0375\text{m}/(\text{s}\sqrt{\text{h}})$ velocity random walk parameters. The artificial noise added to the image coordinates had a standard deviation of 0.1 Pixel. Though the IMU measurement biases are modeled as random walk processes, their values stayed fix for the duration of the simulation. However, the biases were also estimated during the simulation runs, *i.e.* their initial covariance and their process noise power spectral density were initialized with realistic values. The state was initialized with the true values from the reference trajectory after a standstill period of 3.2 seconds. Deviating from the physical model described in sec. 2.3.2 a small amount of pseudo-noise was added to the diagonal elements of the covariance matrix for the landmark position estimates.

The simulation provides ground truth for the position and orientation of the IMU but not for the estimated uncertainty (the covariance matrix). Therefore, the normalized estimation error squared (NEES) is used as a measure of filter consistency for the comparison of the two methods, *cf.* (Bar-Shalom et al., 2001, p. 165):

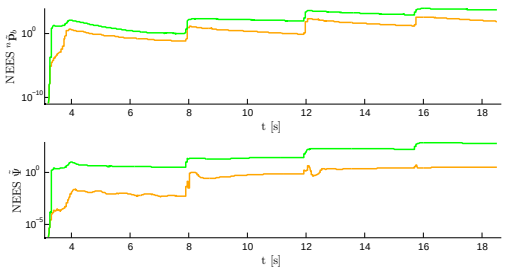
$$NEES\ {}^n\tilde{\mathbf{p}}_b = {}^n\tilde{\mathbf{p}}_b^{\mathrm{T}} P^{-1}_{\tilde{\mathbf{p}},\tilde{\mathbf{p}}}\, {}^n\tilde{\mathbf{p}}_b \tag{27}$$

$$NEES\ \tilde{\Psi} = \tilde{\Psi}^{\mathrm{T}} P^{-1}_{\tilde{\Psi},\tilde{\Psi}}\, \tilde{\Psi} \tag{28}$$

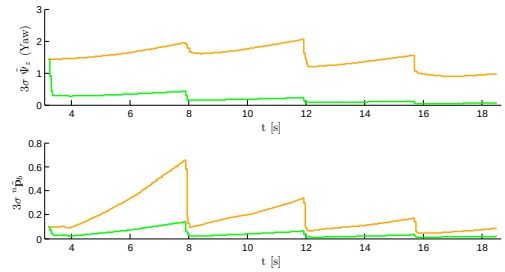
It is also interesting to investigate the covariance bounds for position and orientation errors. Since no measurements are available that relate the sensor system to the navigation frame, aside from the implicitly measured gravity vector, the uncertainty of the position and yaw angle estimates should not fall below their initial estimates.



(a) Simulated hallway trajectory



(b) NEES for position and orientation estimates



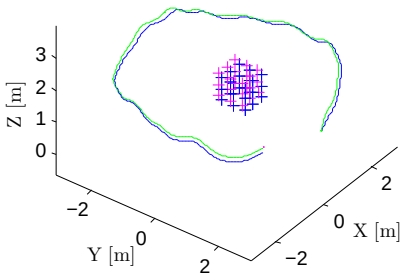
(c) Yaw angle and position covariance bounds

Figure 4: Hallway sequence. Results for one simulation run are shown in Fig. 4(a). Blue: Reference trajectory and reference landmark positions, Pink: Estimated landmark positions, Green: Estimated trajectory. Fig. 4(b) shows the NEES averaged over 25 Monte-Carlo simulation runs. Orange: With cross-covariance estimation for new landmarks. Green: without cross-covariance estimation. Notice the log scale in the NEES-plots. Fig. 4(c) compares the estimated covariance bounds for yaw angle and position estimates. Orange: With cross-covariance estimation for new landmarks. Green: without cross-covariance estimation.

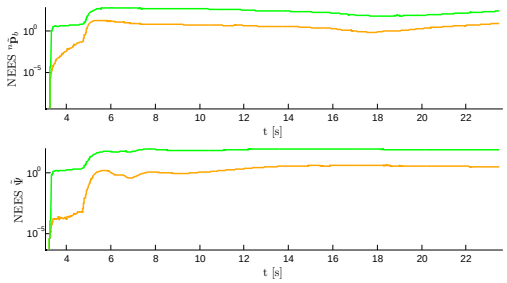
3.2 Results

Fig. 4 presents the results for a simulated trajectory that imitates a walk along a long hallway. During the simulation run, landmarks go out of sight and are replaced by newly initialized landmarks. A comparison of the NEES plots shows that estimating the cross-covariances with the proposed method indeed yields more consistent filter estimates. However, the initialization of new landmarks after 8,12, and 16 seconds goes along with a considerable drop in the uncertainty estimate and an increasing NEES. This is probably because the linearization points used to calculate the derivatives for cross-covariance estimation deviate increasingly from the ground truth during the simulation run.

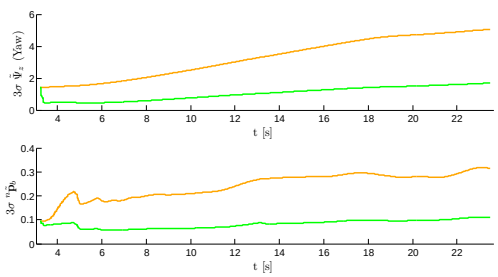
By contrast, fig. 5 shows the evaluation of a trajectory around a cube. Here, the camera's principal axis always points in the direction of the cube so that all landmarks are visible during the whole run, *i.e.* the cube is completely transparent. Thus, landmarks are initialized only once at the beginning of the run when the filter is initialized with the true parameters from the ground truth. Apparently this results in considerable more consistent estimates. In particular, the uncertainty estimate never falls below its initial value when the proposed cross-covariance estimation method is applied.



(a) Simulated trajectory around a cube



(b) NEES for position and orientation estimates



(c) Yaw angle and position covariance bounds

Figure 5: Cube sequence. See fig. 4 for details.

4 CONCLUSIONS

In this work we study the effects of inter-landmark cross-covariance estimation for an EKF-based inertial aided monocular SLAM system. In particular, we describe how the cross-covariances between new features and existing filter state entries may be computed for the special case when the landmarks are parameterized w.r.t. coordinate systems whose position and orientation is also uncertain. This situation naturally arises when parameterizing features with inverse depth polar coordinates w.r.t. the camera in which they were observed for the first time. Using simulation runs, we show that neglecting the cross-covariances for freshly inserted features results in a systematic underestimation of the filter state uncertainty and that this effect may be mitigated with the proposed algorithm.

REFERENCES

Bar-Shalom, Y., Li, X. R. and Kirubarajan, T., 2001. Estimation with Applications to Tracking and Navigation. John Wiley & Sons, Inc.

Civera, J., Davison, A. and Montiel, J., 2007. Inverse depth to depth conversion for monocular slam. In: Proceedings of ICRA.

Dissanayake, G., Newman, P., Clark, S., Durrant-Whyte, H. and Csorba, M., 2001. A solution to the simultaneous localization and map building (slam) problem. IEEE Transactions on Robotics and Automation 17, pp. 229–241.

Farrell, J. and Barth, M., 1999. The Global Positioning System & Inertial Navigation. McGraw-Hill.

Julier, S. and Uhlmann, J., 2001. A counter example to the theory of simultaneous localization and mapping. In: Proceedings of ICRA 2001.

Montiel, J., Civera, J. and Davison, A., 2006. Unified inverse depth parameterization for monocular slam. In: Robotics Science and Systems.

Pietzsch, T., 2008. Efficient feature parameterisation for visual slam using inverse depth bundles. In: Proceedings of BMVC.

Solà, J., 2010. Consistency of the monocular ekf-slam algorithm for three different landmark parameterizations. In: Proceedings of ICRA 2010.

Veth, M. and Raquet, J., 2006. Fusion of low-cost imaging and inertial sensors for navigation. In: Proceedings of the Institute of Navigation GNSS.